

华清远见-成都中心-H5教学部



华清远见-成都中心-H5教学部



目录

什么是DOM

获取文档节点

根据关系找节点及元素节点

节点属性及节点内容

节点的创建、添加、删除、复制和替换

DOM-概述

- DOM , Document Object Model , 文档对象模型 ;
 - document表示窗口中的内容;
 - DOM是表示和操作HTML和XML文档内容的基本API;
 - HTML里面的信息对JS来说都是可以操作的: 增删查改;
- DOM树
 - HTML里面的标签、元素及其内容、注释等, 都可以看做节点; 节点组成DOM树;
 - 父节点: 在一个节点之上的直接节点就是父节点;
 - 子节点: 在一个节点的下一层直接节点就是子节点;
 - 兄弟节点: 同一层上具有相同父节点的节点称为兄弟节点;
 - 后代节点: 在一个节点之下的所有层级节点就是后代节点;
 - 祖先节点: 一个节点的所有上级节点称为祖先节点;

DOM-选取文档节点

- 要想操作文档中的元素，必须先获取到对应的节点；
 - 通过ID获取节点:document对象的getElementById()方法;
 - 只获取一个节点，因为ID是唯一的；
 - 通过名字获取节点：document.getElementsByTagName();
 - 获取所有name属性值是指定值的节点，返回的是类数组的NodeList;
 - 通过标签名获取节点:document.getElementsByTagName();
 - 获取所有的标签是指定标签的节点，返回的是类数组的HTMLCollection;
 - 通过class属性获取节点:document.getElementsByClassName();
 - 获取所有class是指定class的节点，返回的是类数组的HTMLCollection;
 - 通过选择器获取节点:document.querySelectorAll('选择器');
 - querySelectorAll：返回指定选择器对应的所有节点，类数组；
 - 同时支持querySelector：返回指定选择器对应的一个节点；

DOM-根据关系找节点

- 根据节点关系属性；
 - parentNode: 获取节点的父节点，不存在返回null;
 - childNodes: 获取节点的子节点，返回的是类数组(NodeList对象);
 - firstChild: 子节点的第一个节点;
 - lastChild: 子节点的最后一个节点;
 - nextSibling: 该节点的下面那一个兄弟节点;
 - previousSibling: 该节点的上面那一个兄弟节点;

DOM-节点属性

- 节点属性；
 - nodeType: 节点类型；
 - 1--->Element节点；
 - 3--->Text节点；
 - 8--->Comment节点；
 - 9--->Document节点；
 - nodeValue: text节点或comment节点的内容；
 - nodeName: 元素节点的标签名，返回的是标签名的大写；
- document节点属性：
 - document.head: 返回文档的< head>节点；
 - document.body: 返回文档的< body>节点；
 - document.forms: 返回页面中所有表单元素form；
 - document.images: 返回页面所有图片元素（即img标签）；
 - document.links: 返回文档所有设定了href属性的a及area元素；

DOM-根据关系找元素节点

- 根据关系找元素节点，忽略换行、注释、文本等节点；
 - children: 获取节点的子元素节点；
 - firstElementChild: 第一个子元素节点；
 - lastElementChild: 最后一个子元素节点；
 - nextElementSibling: 该节点的下面那一个兄弟元素节点；
 - previousElementSibling: 该节点的上面那一个兄弟元素节点；
 - childElementCount: 子元素节点数量；值和children.length相等；

DOM-元素节点的标签属性操作

- 操作元素节点的属性：
 - attributes: 获取元素节点的标签属性集合;
 - className: 获取或修改元素节点的class;
 - classList: 类数组的方式返回标签的全部class; 该属性常用方法:
 - add(c1, c2...): 增加一个或多个class, 如果已存在就不添加;
 - remove(c1,c2...): 移除一个或多个class, 如果不存在也不会报错;
 - contains(class): 判断类名是否存在, 存在返回TRUE, 否则FALSE;
 - item(index): 返回指定索引位置的class。超出范围返回null;
- 元素节点.id: 获取或修改元素节点的ID ;
- 标准属性及任意属性的操作方法：
 - 系统为标签提供的属性叫标准属性。如img的src, a的href属性等。
 - 对于标准的属性, 我们可以直接进行操作, 如:节点.src = './topimg1.jpg' ;

DOM-元素节点的标签属性操作

- 可以操作任意属性的方法，包括标准属性和自定义属性：
 - `getAttribute('属性名')`：获取属性值；
 - `setAttribute('属性名','属性值')`：设置属性值；
 - `hasAttribute('属性名')`：判断是否有某个属性；
 - `removeAttribute('属性名')`：移除某个属性；
 - 他们的参数都是字符串；
- 有时需要在HTML标签中加一些自己的数据，可以通过自定义属性来实现。
 - W3C为HTML提供的自定义属性的标准方式为：`data-属性名`；
 - 通过元素对象中`dataset`来访问和修改自定义属性；
 - 当然你也可以不按标准方式给标签添加自定义属性；

DOM-节点内容

- 获取或修改内容：
 - innerHTML: 获取一个节点的内容，包括里面的HTML标签，不包含标签本身；
 - outerHTML: 获取一个节点的内容，包括里面的HTML标签，包含标签本身；
 - textContent: 标准的获取元素的纯文本内容；
 - innerText: 也是获取纯文本内容；

DOM-创建及添加节点

- 创建节点：
 - 创建元素节点：document.createElement('标签名');
 - 创建文本节点：document.createTextNode('文本内容');
 - 创建注释节点：document.createComment('注释内容');
- 把节点添加到指定位置：
 - 方式一：节点.appendChild(新节点) 是指在节点的最后插入新节点；
 - 方式二：节点.insertBefore(新节点, 子节点) 是指 在节点的子节点之前插入新节点，如果没有指定子节点，则插到最后；

DOM-节点的删除复制和替换

- 节点的删除、复制和替换：
 - 删除节点：节点.removeChild(指定要删除的子节点);
 - 节点复制：节点.cloneNode(boolean)：参数默认为false;
 - 默认只复制节点自身;
 - boolean为TRUE时表示深层复制：除了复制他自身，还复制子节点;
 - 节点替换：parentNode.replaceChild(newChild, oldChild);
 - 把parentNode的子节点oldChild替换成newChild;

华清远见-成都中心-H5教学部

目录

样式修改操作

几何形状及盒子模型

表单操作

DOM-CSS样式操作

- 改变节点的style属性是最直接的方式；
 - 通过style指定css属性，并改变其值，语法格式是：
节点.style.css属性 = '值';
- CSS修改基本规则
 - 不能访问到用style标签设置的样式；
 - CSS属性名使用小写；
 - CSS属性名如有连字符(-)，则去掉连字符，并将其后的第一个字母大写；
 - CSS属性值必须为字符串，有单位的要带上单位；
- 如果改的属性比较多，推荐使用class进行样式修改；
- 通过改变引用的CSS文件改变样式；

DOM-几何形状

- 查询元素的几何尺寸；
 - 节点.getBoudingClientRect(): 得到一个元素的尺寸和坐标位置；
 - left和top对应的是左上角x,y坐标；
 - right和bottom对应的是右下角x,y坐标；
 - 如果遇到浏览器返回的ClientRect里面如果没有width和height， 可以通过上面的坐标关系计算出来；
- 节点.scrollIntoView(): 让元素在视口中可见；
 - 默认将元素和视口的上边缘对齐；
 - 如果传递参数false， 则将元素的下边缘和视口的下边缘对齐；

DOM-盒子模型

- 得到一个盒子相关尺寸信息属性：
 - offsetWidth: 本身宽度+边框线+左右内边距;
 - offsetHeight: 本身高度+边框线+上下内边距;
 - offsetTop: 相对有定位属性的父节点上偏移量;
 - offsetLeft: 相对有定位属性的父节点左偏移量;
 - clientWidth: 本身的宽度+左右内边距;
 - clientHeight: 本身的高度+上下内边距;
 - clientTop: 上边框线的宽度;
 - clientLeft: 左边框线的宽度;
 - scrollWidth: 盒子的实际宽度(包括不可见部分);
 - scrollHeight: 盒子的实际高度(包括不可见部分);
 - scrollTop: 滚动条向下滚动的距离;
 - scrollLeft: 滚动条向右滚动的距离;
 - window.innerHeight: 浏览器窗口可见区域高度;
 - window.innerWidth: 浏览器窗口可见区域宽度;

DOM-表单操作

- 获取表单和表单元素：
 - `document.forms`: 返回页面中所有表单元素;
 - 表单和表单元素可以用标准的节点获取方式来获取;
- 表单和元素的属性和方法：
 - 表单.`submit()`: 表单提交;
 - 表单.`reset()`: 表单重置;
 - 元素.`type`: 只读, 表单元素类型;
 - 元素.`form`: 只读, 包含该元素的form表单对象, 不存在返回null;
 - 元素.`name`: 只读, 元素的名称;
 - 元素.`value`: 可读可写, 简单理解就是输入框的值;
- 开关按钮：
 - 单选按钮和复选框定义了`checked`属性, 可读可写;
 - 选中时为`true`, 你也可以把他设置为`true`表示选中;

DOM-表单操作

- 表单和元素的事件处理程序：
 - 表单.onsubmit: 监听表单提交事件，表单提交到服务器前，先执行该事件；
 - 如果事件处理程序返回false，则表单提交取消，数据不会提交到服务器处理；
 - 需要注意的是，表单.submit()方法不会触发onsubmit事件；
 - onreset，表单重置事件，处理过程和onsubmit一样；

DOM-表单操作

- select一些常用操作：
 - 节点.options: 返回select所有option选项，类数组的HTMLOptionsCollection;
 - 节点.selectedIndex: 可读可写，被选中的选项索引，下边从0开始;
 - 通过修改selectedIndex的值设置默认选中选项;
 - select节点.value: 得到选中的option值;
 - option节点.text: 得到选项的文本信息，也就是用户看到的选项文字;
 - option节点.value: 得到的是选项的值，和SELECT选中的值有区别;
 - 一样可以通过节点操作对select的option进行增删查改操作;

第30章 事件

华清远见-成都中心-H5教学部



目录

事件基本概念

事件的三个阶段

事件对象

事件绑定方式

事件代理

目录

默认事件阻止

鼠标常见事件

键盘常见事件

防抖与节流

事件-基本概念

- 事件是JS捕捉到的发生在网页上的行为；
- 事件的绑定：
 - 事件源.addEventListener(事件名称, 处理函数);
 - 这种方式使得同一个事件可以绑定多次;
 - IE下绑定事件的方式: 事件源.attachEvent('带on的事件', 处理程序);
- 事件的移除：
 - 事件源.removeEventListener(事件名称, 处理函数);
- 事件的三个阶段
 - 先 捕捉, 后 目标, 再 冒泡;
 - 默认在冒泡的时候执行事件;
 - addEventListener绑定事件, 如果把第三个参数设置为true, 则在捕捉的时候执行事件;
 - 整个事件处理过程, 会有个event事件对象在整个事件过程传播, 是一个全局对象;

事件-基本概念

- 阻止事件传播：`event.stopPropagation()`；
- 阻止浏览器默认事件：`event.preventDefault()`；
 - 比如链接会跳转、右键会出现浏览器的菜单等都是浏览器默认事件； `event.preventDefault()`可以阻止这些事件的发生；
- 事件代理：利用事件冒泡，只指定一个事件处理程序，就可以管理某一类型的所有事件；
 - 可以动态的给节点绑定事件；
 - 监听事件减少，内存及CPU都会降低，运行效率高；
 - 借助`event.target` 或 `event.srcElement`属性实现事件代理；
 - 也称为事件委托；

事件-鼠标常见事件

- 鼠标常见事件：
 - click: 鼠标点击 或 激活 的节点;
 - dblclick: 鼠标双击;
 - mouseover: 鼠标移入;
 - event.relatedTarget返回的是从哪个节点进来的;
 - mouseenter移入事件, 不支持冒泡, 不怎么常用;
 - mouseout: 鼠标移出;
 - event.relatedTarget返回的是出来后即将到哪个节点;
 - mouseleave移出事件, 不支持冒泡, 不怎么常用;
 - mousedown: 鼠标摁下时触发事件;
 - mouseup: 松开;
 - mousemove: 鼠标移动事件;
 - contextmenu: 右键事件;
 - onmousewheel: 鼠标移动事件;

事件-更多事件

- 加载事件：
 - onload: 加载成功事件;
 - onerror : 加载失败事件;
- 滚动条滚动事件：
 - onscroll : 滚动条滚动事件;
- 键盘事件：
 - onkeydown: 键盘摁下事件;
 - onkeyup: 键盘松开事件;
- 输入框事件：
 - onfocus: 获取焦点事件;
 - onblur: 失去焦点事件;

事件-防抖

防抖：触发高频事件后n秒内函数只会执行一次，如果n秒内高频事件再次被触发，则重新计算时间。

思路：每次触发事件时都取消之前的延时调用方法。

节流：高频事件触发，但在n秒内只执行一次，所以节流会稀释函数的执行频率。

思路：每次触发事件时都判断当前是否有等待执行的延时函数。

场景分析：

1. 搜索框input事件，具体使用哪种方案要看业务需求
 - i. 要支持输入实时搜索可使用节流，间隔一段时间就查询相关内容；
 - ii. 要支持输入完成后搜索可使用防抖。
2. 页面resize事件，常见于需要做页面适配，需根据最终呈现的页面情况进行DOM渲染，这种情形一般是使用防抖，因为只需要判断最后一次的变化情况。



海量视频 贴身学习



超多干货 实时更新

THANKS

— 谢谢 —